

AFCRL-71-0211

AD 729451

Analysis and Simulation Branch  
Scientific Problem Library

by

Donald J. Armstrong  
Frances H. Healey

Carolyn M. Mandell  
David Stoll

Analysis & Computer Systems, Inc.  
113 Terrace Hall Avenue  
Burlington, Massachusetts 01803

Contract No. F19628-67-C-0397

Final Report

Period Covered: June 7, 1967 through March 5, 1971

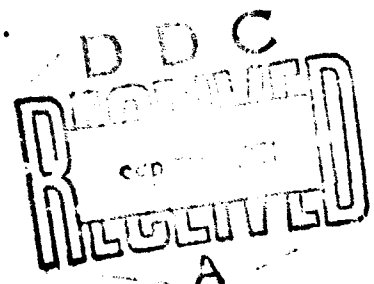
March 1971

Contract Monitor: Austin A. Almon Jr., Analysis and Simulation Branch

This document has been approved for public  
release and sale, its distribution is unlimited.

Prepared  
for

AIR FORCE CAMBRIDGE RESEARCH LABORATORIES  
AIR FORCE SYSTEMS COMMAND  
UNITED STATES AIR FORCE  
BEDFORD, MASSACHUSETTS 01730





AFCRL-71-0211

Analysis and Simulation Branch  
Scientific Problem Library

by

Donald J. Armstrong  
Frances H. Healey

Carolyn M. Mandell  
David Stoll

Analysis & Computer Systems, Inc.  
113 Terrace Hall Avenue  
Burlington, Massachusetts 01803

Contract No. F19628-67-C-0397

Final Report

Period Covered: June 7, 1967 through March 5, 1971

March 1971

Contract Monitor: Austin A. Almon Jr., Analysis and Simulation Branch

This document has been approved for public  
release and sale, its distribution is unlimited.

Prepared  
for

AIR FORCE CAMBRIDGE RESEARCH LABORATORIES  
AIR FORCE SYSTEMS COMMAND  
UNITED STATES AIR FORCE  
BEDFORD, MASSACHUSETTS 01730

Unclassified

Security Classification

## DOCUMENT CONTROL DATA - R &amp; D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author) Analysis & Computer Systems, Inc. 113 Terrace Hall Avenue Burlington, Massachusetts 01803		2a. REPORT SECURITY CLASSIFICATION Unclassified	
		2b. GROUP	
3. REPORT TITLE ANALYSIS AND SIMULATION BRANCH SCIENTIFIC PROBLEM LIBRARY			
4. DESCRIPTIVE NOTES (Type of report and inclusive dates) Scientific. Final. 7 June 1967 - 5 March 1971      Approved 30 June 1971			
5. AUTHOR(S) (First name, middle initial, last name) Donald J. Armstrong      Carolynne M. Mandell Frances H. Healey      David Stoll			
6. REPORT DATE March 1971		7a. TOTAL NO. OF PAGES 67	7b. NO. OF REFS 0
8a. CONTRACT OR GRANT NO. F19628-67-C-0397		9a. ORIGINATOR'S REPORT NUMBER(S)	
b. PROJECT NO. Task, Work Unit Nos. N/A      N/A      N/A			
c. DoD Element: 61102F		9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)	
d. DoD Subelement: 681300		AFCRL-71-0211	
10. DISTRIBUTION STATEMENT 1- This document has been approved for public release and sale; its distribution is unlimited.			
11. SUPPLEMENTARY NOTES TECH, OTHER		12. SPONSORING MILITARY ACTIVITY Air Force Cambridge Research Laboratories (SUY), L.G. Hanscom Field Bedford, Massachusetts 01730	
13. ABSTRACT <p>This report is the final report prepared to record the status and progress of the Scientific Problem Library of the Analysis and Simulation Branch (SUYA). This report describes the efforts expended in the establishment and maintenance of the library of completed computer programs, the establishment of procedures for formally accepting and testing computer programs prior to their storage in the library system and the investigation of new techniques for automatic storage and retrieval of information in the program library.</p>			

DD FORM 1 NOV 65 1473

Unclassified

Security Classification

Unclassified

Security Classification

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Problem Library Standardization Program Verification Documentation Problem Number Abstracts Category Storage Retrieval Update Management Reports Statistical Analysis						

Unclassified

Security Classification

## ABSTRACT

This report is the final report prepared to record the status and progress of the Scientific Problem Library of the Analysis and Simulation Branch (SUYA). This report describes the efforts expended in the establishment and maintenance of the library of completed computer programs, the establishment of procedures for formally accepting and testing computer programs prior to their storage in the library system and the investigation of new techniques for automatic storage and retrieval of information in the program library.

## TABLE OF CONTENTS

<u>SECTION</u>		<u>PAGE</u>
	ABSTRACT	i
	LIST OF ILLUSTRATIONS	v
	FOREWORD	vii
1	INTRODUCTION	1-1
2	PROGRAM DOCUMENTATION AND LIBRARY SYSTEM (PDALS) DESIGN	2-1
3	DOCUMENTATION	3-1
4	LIBRARY FILE SYSTEM CLASSIFICATION	4-1
5	PROGRAM STORAGE	5-1
6	INFORMATION RETRIEVAL	6-1
7	CONVERSION	7-1
8	CONTRIBUTORS	8-1
9	APPENDIX A	9-1
10	APPENDIX B	10-1

## LIST OF ILLUSTRATIONS

<u>FIGURE</u>		<u>PAGE</u>
1	Program Documentation and Library System	2-3
1-A	Documentation Verification	2-4
1-B	Classification and Storage of Completed Programs	2-5
1-C	Program Deck Request	2-6
1-D	PLIF Information Request (Statistical Reports)	2-7



## FOREWORD

The design of the Scientific Problem Library, together with the computer systems that implement these designs, are the result of analytical research performed for,

The Analysis and Simulation Branch (SUYA)  
AFCRL Computation Center  
Air Force Cambridge Research Laboratories  
Bedford, Massachusetts 01730

The computer programs contained in this report may be obtained from the above organization upon request by referencing Problem Number 1083.

## SECTION 1

### INTRODUCTION

On June 6, 1967, Analysis & Computer Systems, Inc. was awarded a contract to investigate, analyze and design a totally integrated complex program documentation system. The proposed task was as follows:

Perform the required research and analysis for the establishment of a program library of analytical routines in support of the pure and applied scientific investigations conducted by the Analysis & Simulation Branch at AFCRL. Such investigations shall include magnetic tape, microfilm or any other storage media deemed feasible and will consist of, but not be limited to the following:

Develop and implement scientific documentation procedures for acceptance of completed mathematical and analytical routines designed and developed for the Data Analysis Branch in support of research projects at AFCRL.

Investigate methods and develop new techniques to classify, store, maintain, and update scientific project problem studies for immediate and timely retrieval upon request by any user from the Analysis & Simulation Branch.

Investigate methods and develop optimum techniques for miniscule storage media utilizing constant and partial update of documentation for large satellite, orbital and laboratory data reduction and analytical continuation projects.

Provide consulting and maintenance operation assistance to the Analysis & Simulation Branch Library of Scientific Computer Programs upon request of the Contracting Officer.

In December 1968, Analysis & Computer Systems, Inc. received an additional task which expanded the task as follows:

Study and make provisions in the existing program library documentation retrieval system to incorporate documentation produced by the analog-hybrid analytical problems.

Develop a system of reports from the established scientific library documentation system for use as management analysis tools. This system shall include, but not be limited to, program scheduling, evaluation, performance, cost analysis and automated procedures for status reporting and problem schedule monitoring.

Prepare abstracts of the programs in the AFCRL (SUYA) Magnetic Tape Library for timely and continuous periodic distribution to interested researchers within the AFCRL scientific community.

Perform the required research and development to implement the MARC II format as it will apply to the AFCL Research Library Information and Retrieval System.

This report documents the efforts under subject contract.

## SECTION 2

### PROGRAM DOCUMENTATION AND LIBRARY SYSTEM (PDALS) DESIGN

In order to design a library system of computer programs, it was necessary to:

1. Investigate the various types of data to be included in the library.
2. Become knowledgeable of the computer on which the library system was to operate (i.e., IBM 7094 II/7044 DCS).
3. Become aware of both management and library users' needs and requirements.
4. Study and select the techniques necessary to satisfy these needs.

In the ensuing study, the following tasks were recommended to, and approved by, government technical monitors as important parts of the library system to be developed.

a. The development of documentation standards and means of testing their accuracy and validity. It is essential that every piece of pertinent information be made available for use by an independent source, and it is equally essential that this information be clear, accurate and complete.

b. The development of techniques to store, maintain, update, and retrieve the completed scientific programs and/or data. It is necessary that these techniques be capable of handling large volumes of data and, at the same time, be compatible with the computer system available.

c. The means for disseminating titles, abstracts, authors and other pertinent information concerning completed problems. This type of information can be meaningful in a number of ways; for example, it provides a means for users to ensure against duplication of effort, a means of exchanging ideas with researchers on similar projects and another means of keeping abreast of the state-of-the-art.

d. The means for providing a system of reports for statistical analysis of usage. This provides management with a tool for program scheduling, performance evaluation and cost analysis. It also ensures that the problem library system is operating efficiently.

Figure 1 (Program Documentation and Library System (PDALS)) shows the general overall design of the scientific problem library as it evolved from a thorough study and the analysis and design which was necessary to complete the task. It is also a general flow diagram of the system as it presently operates.

Circled numbers on Figures 1 to 1-D are included as items of reference for the technical discussion that follows.

The following sections will describe the various functions performed in the development of the system and in its everyday operation. Included in these sections will be ideas to improve upon the existing library system considering the additional speed, memory size, safety features, and overall efficiency of the CDC 6600 system.

PROGRAM DOCUMENTATION AND LIBRARY SYSTEM

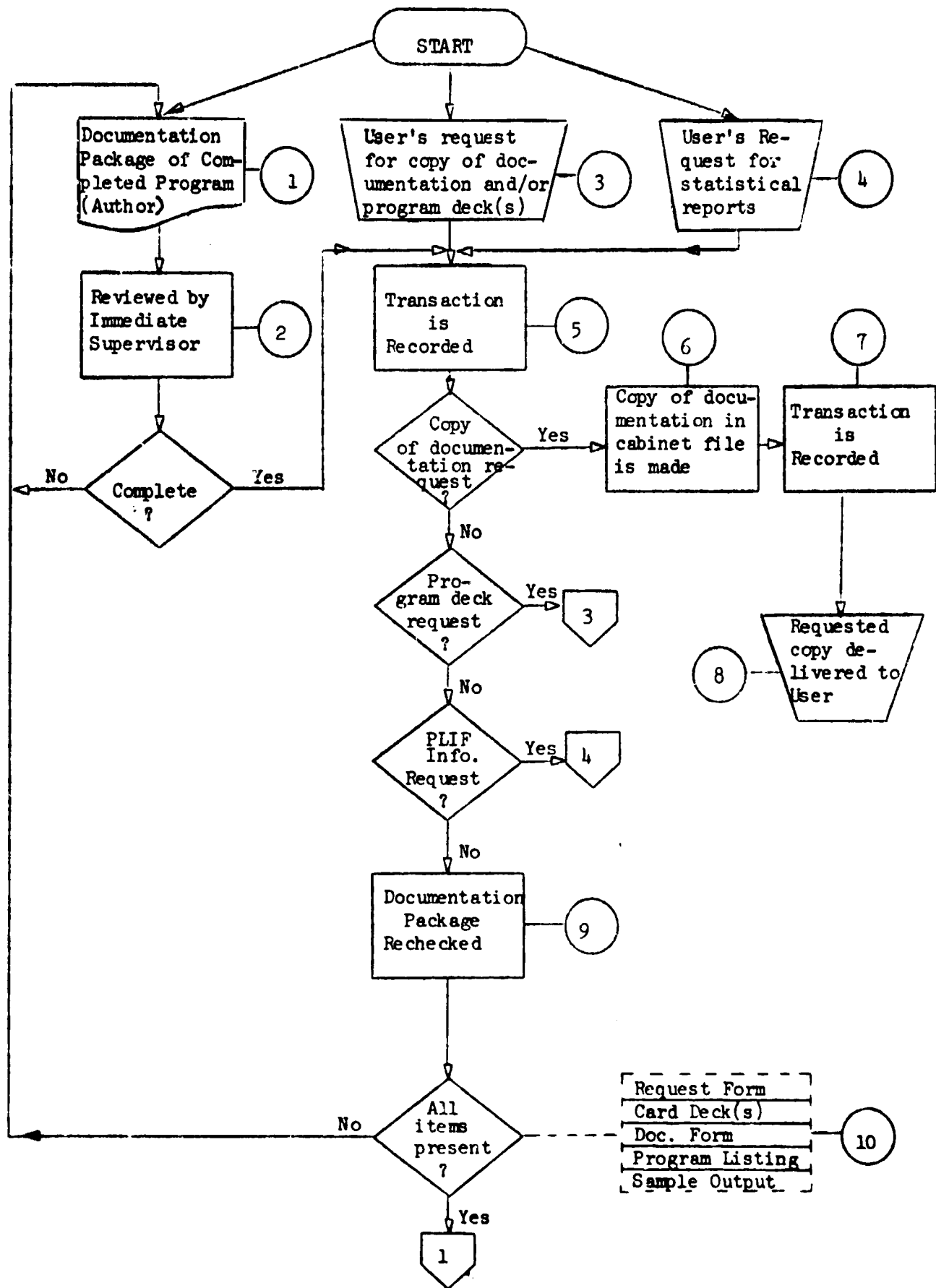


Figure 1

# DOCUMENTATION VERIFICATION

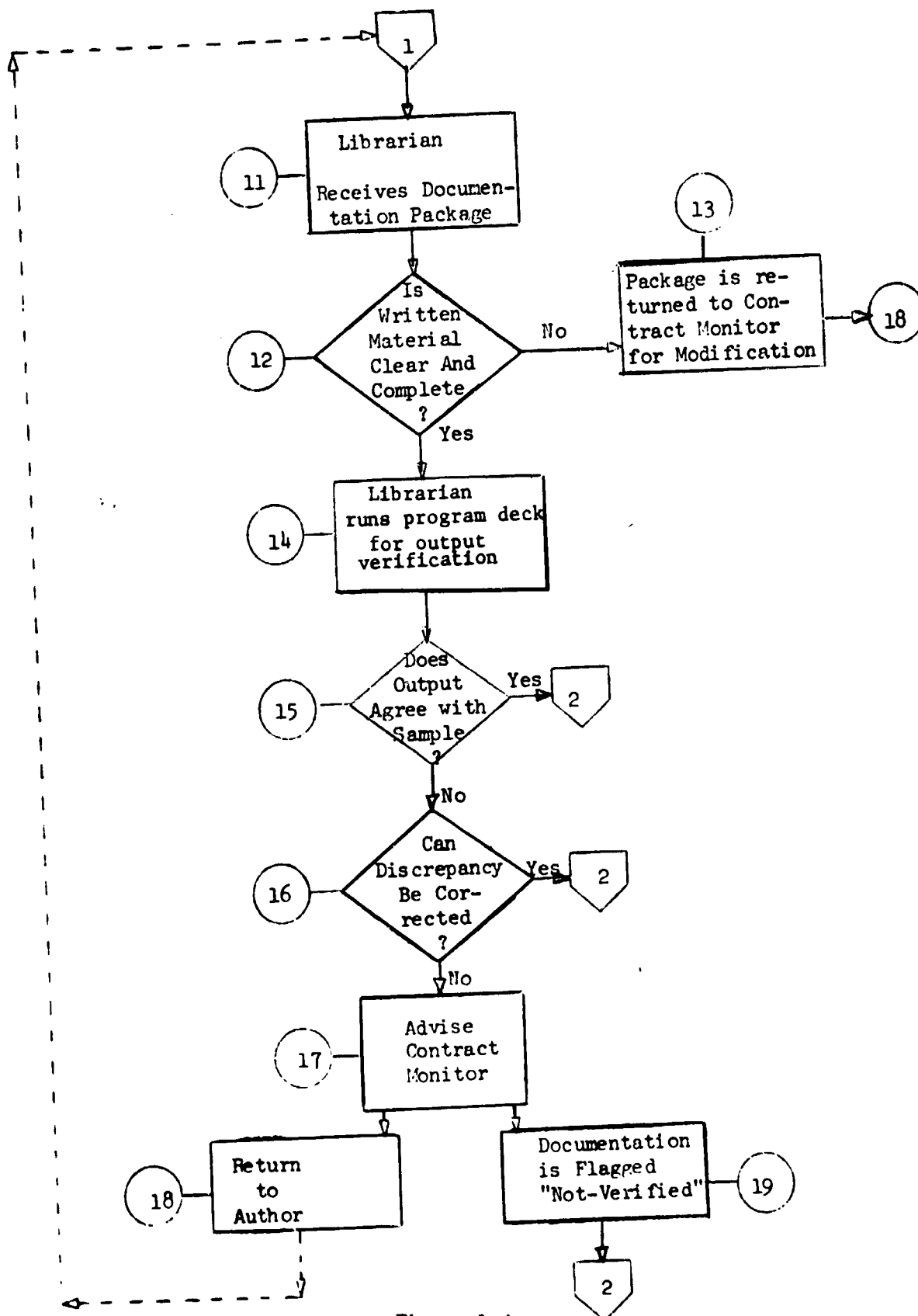


Figure 1-A

CLASSIFICATION AND STORAGE  
OF COMPLETED PROGRAMS

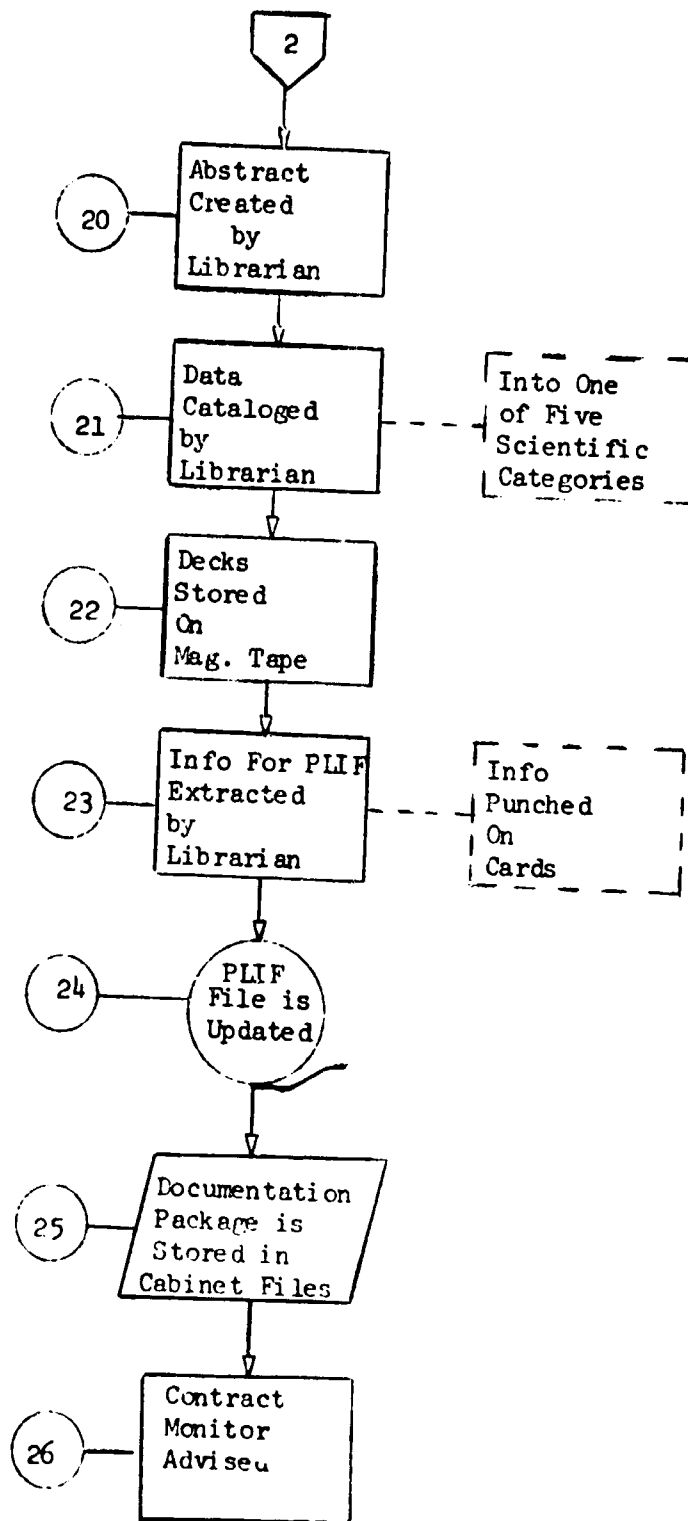


Figure 1-B



PROGRAM DECK REQUEST

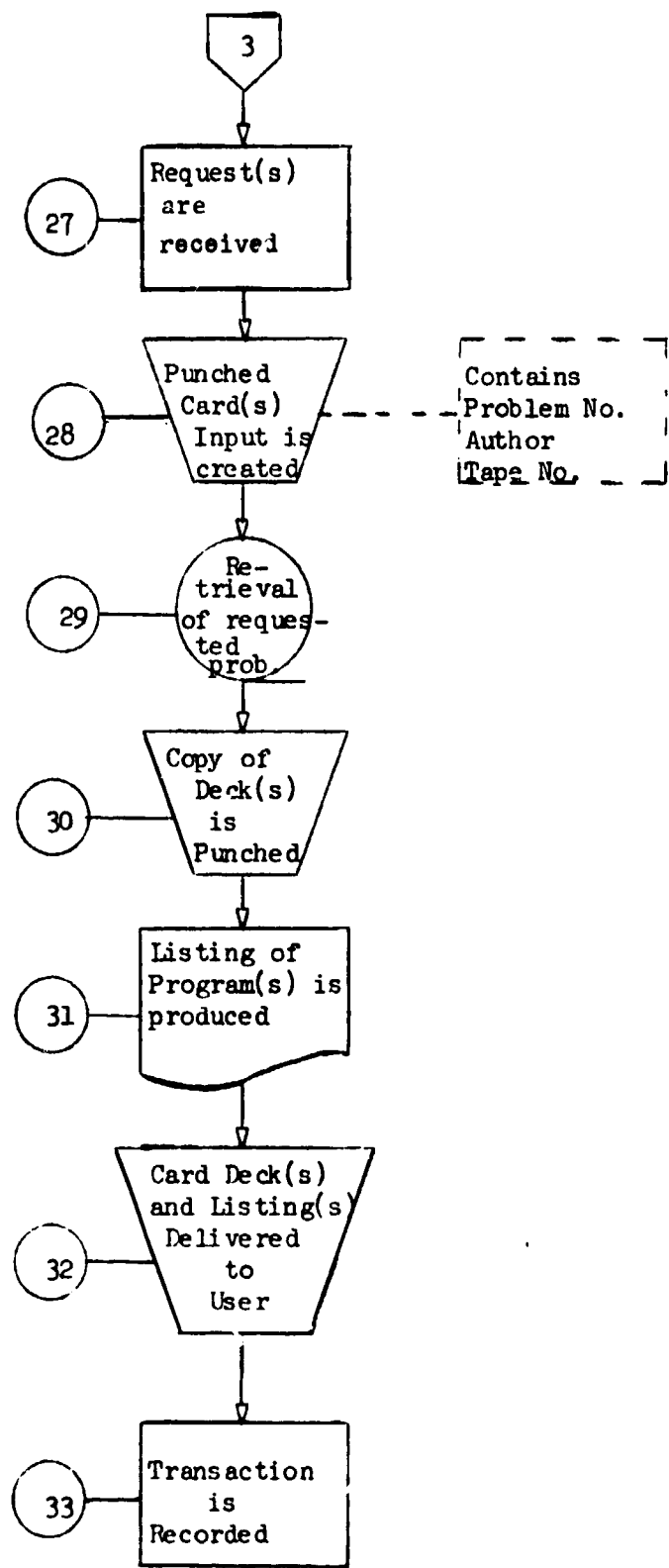


Figure 1-C

PLIF INFORMATION REQUEST  
(STATISTICAL REPORTS)

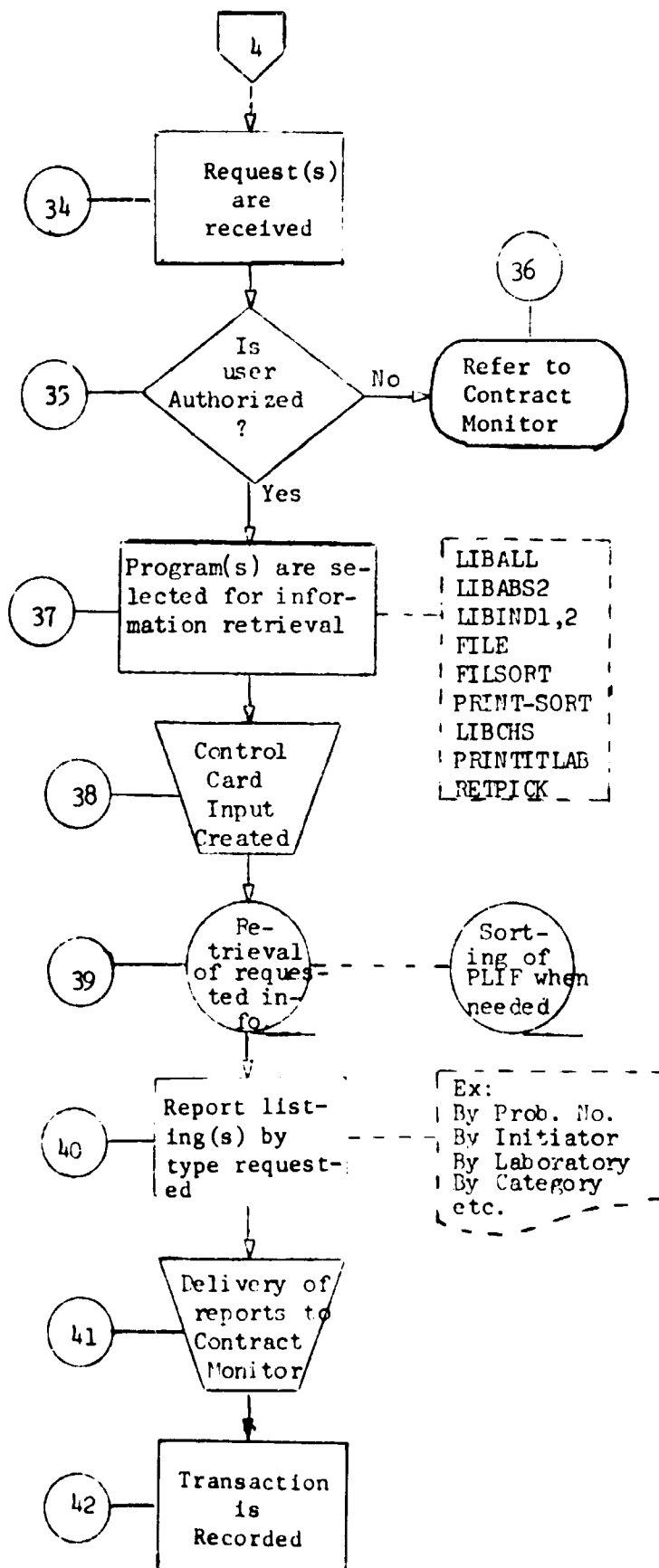


Figure 1-D

### SECTION 3

#### DOCUMENTATION

Standards for a documentation package are a function of the use for which the documentation is intended. As was previously mentioned, every possible piece of information must be included in the documentation so that not only a history of the program is maintained to aid the author in any future modifications and/or additions that may occur, but also a means is provided for someone other than the author to learn and take over the computer program if need arises.

The following elements were proposed and accepted as a documentation package for each scientific computer program completed for the Analysis and Simulation Branch of the AFCRL Computation Center.

1. Card deck(s) - a working copy of the computer program card deck(s). This must contain all of the source language decks and should also include the object decks (binary) when available. Sample data input to the program(s) must be made available for verification of operation.

2. Program Listing - a listing of all computer instructions both of compiler language nature and machine language nature that are included in the computer program.

3. Request form - a copy of the original program request for computer programming. This sheet supplies the following information:

- a. Problem number assigned to this task.
- b. Laboratory and Branch of initiator.
- c. Authorized initiator.
- d. Brief description of task.
- e. Estimated computer production frequency.
- f. Estimated computer workload estimate.
- g. Estimated computer problem duration.
- h. Programmer's name.
- i. Project and task number.
- j. Laboratory authorization signature.

4. Abstract - Typed bibliographic data describing the scientific work involved and the computer work necessary to complete the assigned task.

5. Documentation form - a precise written description of each computer program completed under the particular problem number assigned. Shown in the Appendix of this report is the form developed by A.C.S.I. personnel for use while the DCS was operational at AFCRL as well as the proposed form to be used for the CDC 6600.

6. Sample output - listings of sample results for each program submitted including standard computer print listings, digital display

plots, generated paper tape and/or sample format of a record on a magnetic tape generated by the program.

It is imperative that strict adherence to these documentation standards be maintained for an efficient computer program library system. With this in mind, the following procedures for verification of documentation were proposed and they are the procedures that are presently followed.

#### VERIFICATION OF DOCUMENTATION

Since the documentation package is the sole output of extensive programming time and effort, ACSI personnel realized the importance of maintaining a number of checkpoints to verify the accuracy of the package and this aspect is shown in items 2, 9, 14 and 15 of Figures 1 and 1-A.

All material for a particular completed program(s) is submitted to his supervisor by the individual who performed the work. It is the responsibility of this supervisor to examine the material for clarity, completeness and adequacy of technical content (see item 2, Figure 1). If the package lacks any of the items or is unclear, the documentation is returned to the author. When he is satisfied that all elements of the documentation package, as outlined in Part C of the technical approach, are present and complete, the package is submitted to the Analysis and Simulation Branch personnel. Again each item is checked (see item 9, Figure 1) and a folder is prepared for that particular problem number to record its status and location. If all items are present, the documentation package is given to the librarian (see item 11, Figure 1-A). It is the librarian's responsibility to verify in detail the written material contained in the documentation form as described previously in this report (see item 12, Figure 1-A).

At the bottom of each page in the documentation form are guidelines designed to aid the author in completing each of the sections. The librarian must examine each section to ensure that:

1. The correct information is presented relative to the particular subject heading.
2. Sufficient information is available to describe what was done and how it was done.
3. The information is prepared in a form easily understandable by anyone having no prior knowledge of the task.

If the librarian finds the written material to be unclear or incomplete, the documentation is returned to the author for modification.

Another phase of the documentation verification is the operation of the program to verify results (see item 14, Figure 1-A). Since sample input and a sample output listing are included in the package, it remains for the librarian to submit the source deck, and the object deck when available, for a computer run, and to compare the results with the sample output listing for verification. If there is any discrepancy in the output received, the contract monitor is advised (see item 17, Figure 1-A) for further action. In some cases the discrepancy is unavoidable (for example, an original data tape is destroyed), in which case the contract monitor is advised that the documentation be flagged "not-verified" (see item 19, Figure 1-A). He may also elect to return the package to the contract supervisor for correction (see item 18, Figure 1-A).

If all tests for the documentation verification are passed, the task remains to classify, store and record the program package that is to be added to the library file of completed problems. The next section will describe in detail the steps taken in this direction as outlined on Figure 1 of the "PDALS" design.

## SECTION 4

### LIBRARY FILE SYSTEM CLASSIFICATION

As was previously mentioned in the introduction, one of the requirements of contract F19628-67-C-0397 was to investigate methods and develop new techniques to classify, store, maintain, and update scientific Project studies for immediate and timely retrieval upon request by any user from the Data Analysis Branch (presently the Analysis and Simulation Branch). In order to classify programs, one must have experience in the various scientific areas being supported by Analysis and Simulation Branch personnel and also be capable of determining from the functional description of the documentation form the particular scientific area to which the program relates (see item 21, Figure 1-B). The following five categories were recommended and approved as general categories for classifying completed programs:

- a. Mathematical Analysis
- b. Orbital Determination and Operations
- c. Analog and Hybrid Simulation
- d. Non-Numeric Systems
- e. Rocket and Satellite Analysis

The following list of scientific areas supported by work performed by the Analysis and Simulation Branch and the category with which each is associated is presented as a guideline in classifying each completed program:

- a. Mathematical Analysis
  1. Development of numerical and analytical techniques
  2. Statistical analysis
  3. Power spectral density studies
  4. Solution of ordinary and partial differential equations
  5. Eigenfunctions, Eigenvalues
  6. Integral equations and functions
  7. Development of on-line numerical techniques
  8. Least-squares fit
  9. Differencing techniques
  10. Mathematical contouring
  11. Monte-Carlo technique
  12. Multiple regression techniques
  13. Mathematical modeling
- b. Orbital Determination and Operations
  1. Precision ephemerides of artificial and natural satellites
  2. Command coverage for satellite vehicle and rocket flights
  3. View periods for satellite vehicles and rocket flights
  4. Solar-lunar orbital investigations
  5. Eclipse analysis

6. Pre-launch design studies
  7. Triangulation and laser viewing periods
  8. Archiving of interagency orbital data
  9. Space vehicle characteristic archiving
  10. Atmospheric density modelling as pertinent to precision ephemerides
  11. Geodetic constant investigation as pertinent to precision ephemerides
  12. General aspect investigations
- c. Analog and Hybrid Simulation
1. Development of mathematical model for simulation of physical systems
  2. Solution of nonlinear systems
  3. Development of pure analog and hybrid techniques
  4. Vehicle design studies (balloon, satellite, rocket components, et al.)
- d. Non-Numeric Systems (Graphic and Reduction Techniques)
1. Development of large analytical systems from raw data to effective scientific analysis
  2. On-line development of graphic techniques (contour mapping, etc.)
  3. Balloon project data reduction
  4. Pattern recognition
  5. Scintillation studies
  6. In-house laboratory data reduction efforts
  7. Radar data reduction studies
  8. Radiosonde data reduction
  9. Mapping simulations
    - i. Digital planimeter studies
    - ii. Wind profile generation
  10. Information storage and retrieval systems
  11. AFCRL and SUYA library efforts
  12. Program and systems conversions
- e. Rocket and Satellite Analysis
1. Analysis of satellite experiments for DOD and NASA vehicles
  2. Rocket trajectories
  3. Data Reduction from a rocket flight
  4. Satellite data reduction

Subclassification of these categories is envisioned as the library base becomes more extensive.

## SECTION 5

### PROGRAM STORAGE

#### 5.1 Card Deck File

When the documentation has been completed and verified and the program classified into one of the five categories, the problem of storing all of the program data remains. The computer program "LIBTAPE" was written to store the program card decks (both source and object levels) onto magnetic tape. Prior to running "LIBTAPE" however, all of the system control cards for each program deck are changed to comment cards so as not to interfere with the DCS operating system. Input cards must be key-punched to preface each of the decks to be stored reflecting the problem number, deck name, and the modification number, if any. The entire deck is then submitted to be run, using program "LIBTAPE", with a magnetic tape containing previously completed program decks (see item 22, Figure 1-B). A subroutine of "LIBTAPE" written in machine language determines whether the card being read is of source or object origin. A record of this transaction is maintained by punching and storing a card containing the following information:

1. Date decks were stored on tape
2. Problem Number
3. Project Number
4. Program Name
5. Initiator (Researcher)
6. Author (Programmer)
7. Tape Name.

#### 5.2 PLIF (Program Library Information File)

The PLIF was designed to provide management with a tool for program scheduling, evaluation of performance and cost analysis. The PLIF is a collection of pertinent data for each completed problem number representing operational computer programs written by and/or for the Analysis and Simulation Branch at AFCRL. The PLIF is created and updated periodically using program "LIBABS".

After extensive study into the needs of management and other users, it was decided that the following information be extracted from the documentation package and keypunched onto computer cards for subsequent storage on magnetic tape (see item 23, Figure 1-B):

1. Problem Number
2. Project Number
3. Initiator
4. Lab or Branch of Initiator
5. Author and Company
6. Hardware
7. Software



8. Category
9. Title
10. Title Description
11. Indication of Status (verified or not verified)
12. Magnetic Tape Name (where source decks are stored)
13. Date of Issue (of Problem Number).

During the early stages of creating the PLIF, it was found that existing documentation, written prior to the adoption of the documentation standards as outlined in this report, could not provide enough of the information required. The task of upgrading existing documentation was undertaken. Existing documentation packages were returned to their authors, if still employed at AFCRL at the time, for upgrading into the new format. Those packages which could not be returned were rebuilt and reformatted. Out of the 168 packages, 61 were considered complete and acceptable after they were reformatted, but the remaining packages had to be resolved on an individual basis. As of this writing, there are 669 completed problem numbers entered into the PLIF.

The following computer programs were written to enter the data onto the PLIF.

Program "LIBABS" was written to store on magnetic tape all of the aforementioned information. A file is created on tape for each problem number submitted. A "file", when used in reference to problem numbers, is interpreted to mean a group of records which make up one problem number and should not be interpreted to mean a physical file with an end-file tape mark. Since the title description for each job varies in the number of punched cards required for storage, the number of records needed to describe each problem number also varies. Program "LIBABS" stores all of the problem number files on tape in numerical order (beginning with number 1001). If the PLIF magnetic tape and its reserve copy were destroyed, program "LIBABS" could be used to recreate it. The program also outputs a print listing of the problem number files in numerical order, two to a page, from number 1001 to the most recent problem recorded on the PLIF. A list of the categories is also printed at the beginning of the listing to aid the user in relating the category number of a particular problem number to its descriptive heading.

Program "INSERT-ABS" was written for ease in modifying the existing listing of program library information files. This program allows the librarian to update the PLIF by either inserting new files and/or changing any of the fields in the files. Control cards must be punched to direct the program to add, modify, or delete the information contained on subsequent cards. The tape created by program "LIBABS" is updated accordingly and a new listing is produced in the same format as that of "LIBABS".

### 5.3 Abstract File

In the "Documentation" section, the term ABSTRACT was listed as an item to be included in the documentation package. The ABSTRACT contains

bibliographic information describing the scientific work involved including pertinent facts about the computer program itself. This information is usually too lengthy to be stored on the PLIF; therefore, a general title description is stored on tape while the ABSTRACT becomes a manual operation and is maintained as a cross reference to the PLIF.

In order to construct the ABSTRACT, the librarian must familiarize herself with the technical problem and also develop expertise in extracting from the introduction, functional description and output portions of the documentation form the essential information. The ABSTRACT is typed and subsequently filed (see item 20, Figure 1-B).

Future works should include a study to further circulate these computer program abstracts. This study would encompass, but not be restricted to, an investigation into the requirements necessary to publish a journal of computer abstracts on a regular (i.e. quarterly) basis by the Analysis and Simulation Branch at AFCRL if requested. This journal could be made available to interested parties on either a subscription or an individual basis. The journal should include program titles and abstracts arranged by category. Various indices could be included to aid in locating any particular program. For example, some of the retrieval programs could provide a subject index, an initiator and branch index, an equipment requirement index (presently encompassing the IBM 7044, 1460, 7094 II/7044 DCS, 360 series, DEC-PDP series and the CDC 3000 and 6000 series) and a key-word index.

For each program listed, information, such as the computer language in which the program was written, the number of cards comprising the program, and the computer used to check out the program would be made available. This journal would provide researchers at AFCRL and the scientific community in general another means of keeping abreast of the state-of-the art.

#### 5.4 Cabinet File

All documentation, source decks, and transaction records are stored with the Analysis and Simulation Branch, AFCRL Computation Center.

As the documents are now becoming voluminous and storage may become a problem as the library continues to increase in size, it is proposed that methods and equipment available, such as microfilm devices, be investigated for possible use to reduce the size of the storage required.

#### 5.5 Maintenance and Update

In any system as large as the problem library system, certain maintenance procedures are mandatory. Oftentimes tapes containing either source decks or information files are damaged and must be recreated. Periodic

computer runs must be made to ensure that certain problems have been completed and are available for retrieval. Management reports must be obtained upon request. Transactions that occur during the many phases of the library system must be recorded and often referred to. Source decks and/or object decks must be retrieved for the user upon request, and copies must be made of documentation. Equally important is the task of updating the different files in the system. As soon as a job has been completed, it must be made available for use. This means that the card deck file, PLIF, abstract file and cabinet file must be updated weekly to ensure current and accurate files.

## SECTION 6

### INFORMATION RETRIEVAL

Whenever a request is made by a user for a program deck to be reproduced and listed (see item 3, Figure 1), program "LIBPUN" is used to search the tape containing the source or object decks of the particular problem number(s) and punch out the deck(s) associated with that problem number(s) (see item 30, Figure 1-C). Program "LIBPUN" also produces a print listing of the source deck(s) contained within the problem number (see item 31, Figure 1-C). Control cards are prepared (see item 28, Figure 1-C) containing the problem number requested, the initiator and the tape name. This function will be continued, but in addition, safeguards against access to this file by unauthorized users through adaptations of the "KEYWORD" feature offered by the CDC 6600 computer system will be proposed.

#### 6.1 Documentation

Whenever a request is made by a user for a copy of the documentation (see item 3, Figure 1), the folder is retrieved from the cabinet file, a copy obtained of the requested information (see item 6, Figure 1), the transaction recorded and the copy delivered to the user (see items 7, 8, Figure 1).

#### 6.2 Management Reports

Perhaps the most important output of the PDAL system is the management reports. These reports are designed to aid management in program scheduling, performance analysis and cost analysis. They also reflect the efficiency of the PDAL system itself so that periodic reports assist in the maintenance phase of the operation. Several programs provide these reports.

Program "SMPG" was written to list the information cards for every problem number issued by AFCRL beginning with number 1001. These cards are in groups of three per problem number and are prepared whenever a problem number is issued. This program is a means of providing a convenient reference to all problem numbers, completed or in process. The information contained on these cards for each problem number consists of the number itself, project number, description of the work being performed, initiator's name and branch, programmer's name and company, category, date the problem number was issued, and in the most recent problems, the estimated completion date. These cards are listed with no change in format, except for the omission of superfluous characters. This summarized listing provides a quick-look at the status of all problem numbers.

Program "LIBALL" was written to combine the completed problem numbers in the program library information listing with the problem numbers which are still in process, and to list selected information concerning them under the heading "SUYA Problem Number Status Listing". The information is extracted from the master tape generated by programs "LIBABS" and "INSERT-ABS" and from the AFCRL punched cards. The information extracted consists of: the Problem Number, Project Number, Initiator, Lab, Author, Company, Category, Date Opened, Estimated Completion Date, and Problem Status. The Status data will be flagged "completed" if the information originated from tape, and left blank if from cards. In the event there is a gap in sequence, either of

two subroutines will write the missing problem number. "NCRD" is called before the program writes information from cards, and "JCRD" is called before the program writes from tape, in order to check for missing problem numbers. There is an option of printing the total number of problem numbers issued to a particular company. All listings are prefaced by categories and their definitions and are concluded with a count of the completed problem numbers.

Program "LIBABS2" was written to afford the option of listing either all of the problem number files or selected ones designated by input control cards. If all of the files are required, the word "ALL" is punched on the second data card and the program prints out all the problem numbers contained on the master output tape from programs "LIBABS" and "INSERT-ABS". If the word "SOME" is punched, it is necessary to indicate the particular problem numbers desired. The categories and their definitions are printed at the beginning of the listing.

Program "LIBIND1" is the first of two programs written to create a key-word index of the titles for each job in the Program Library Information List (PLIF). The input tape used is created by either "LIBABS" or "INSERT-ABS", and the output tape is created for input to program "LIBIND2". The initial phase of the formation of the key-word index consists of writing the titles and their related problem numbers onto an intermediate tape. This tape is rewound and each word of each title is checked for beginning letter "A". The words beginning with "A" are listed, followed by the remainder of the title and the problem number. Once the intermediate tape of titles has been completed using this process, it is rewound and the process is repeated for "B", "C", ... "Z", and digits "0" through "9". The result is a listing and output tape of titles and their problem numbers chronologically arranged by each problem number. Each title is listed within each alphabetic grouping as many times as a word appears with the first letter belonging to that particular group. For example, many titles may appear having their first word "IONOSPHERE".

Program "LIBIND2" was written to provide the option of removing insignificant words from a title for display purposes. The significant words of the title are sorted alphabetically, printed, and stored on an output tape. The format of "LIBIND2" is similar to that of "LIBIND1" except that the listing of key words is alphabetized and all words such as conjunctions and prepositions are ignored.

As was previously mentioned in the classification and abstract file parts of this report, the two programs "LIBIND1" and "LIBIND2" can be utilized to provide automated classification techniques and also be used to aid in the preparation of a computer program abstract journal.

The following programs were written to yield optimum use of the PLIF and to provide management with a series of reports arranged in various selected forms.

Program "FILE" was written to convert the PLIF files to record sizes compatible with the CDC sort system. This modification of the record size is made to accommodate the sort routine, which cannot recognize record lengths in the same manner as the Fortran system. The input tape created by "LIBABS" or "INSERT-ABS" contains one "file" for each problem number. Each file containing library documentation information is of a different length, because the title descriptions vary in length for each job. To sort a particular field in each record requires a fixed position in that record. Program "FILE" reads the master tape prepared by programs "LIBABS" and "INSERT-ABS", extracts the fields to be sorted with the rest of the file and packs the remainder of the record with blanks to fix each record length at 345 words. The newly created tape is input to "FILSORT", the sort routine. The total number of problem number files that have been selected for sorting is printed.

Program "FILSORT" is an adaptation of the CDC system sort routine to the particular requirements of the program library information data to be sorted. The problem number files, having been prepared by program "FILE", are sorted alphabetically by either initiator's name or lab, or numerically by problem number or project number. This information which is normally located in the last record of each problem number "file" has been relocated at the beginning of the record of 345 words by program "FILE". In order to choose any particular field to be sorted, it is necessary to know the exact location of the problem number or project number or initiator or lab field on the input tape. These fields are indicated on a punched card contained in the program deck for "FILSORT". Another card is punched indicating the format to be used according to the item selected for sorting. The output is a newly created tape of all problem library information sorted by one of the four options previously mentioned. This tape is input to "PRINT-SORT" and "LIBCHS".

Program "PRINT-SORT" reads the tape created by "FILSORT" and outputs a printed listing of the program library information which has been sorted either alphabetically by initiator, branch, problem number or project number. The format of the listing is the same as that used in "LIBABS" and "INSERT-ABS".

Program "LIBCHS" was written to produce listings of the problem library information according to one of the following major sorts:

- a. Problem Number
- b. Project Number
- c. Branch
- d. Initiator

In addition, each of the above has the option of selection:

- a. All completed problem numbers (not necessarily verified) or
- b. All verified problem numbers.

Program "PRINTITLAB" was written to produce a listing of titles along with their title descriptions sorted by problem number.

All of the previously mentioned programs were written to provide management with various types of reports concerning all of the problem numbers available in the library system. The following program was written to provide management with listings concerning selected files contained in the PLIF.

Program "RETPICK" produces listings according to one of the following parameters input to the program:

- a. Tape Number
- b. Project Number
- c. Category
- d. Program Initiator
- e. Author
- f. Documentation
- g. Hardware
- h. Software
- i. Branch
- j. Company Name

One input card at a time is read and checked against the corresponding field in the file. When a match is made, the file is printed. Any number of cards may be processed, each containing one of the parameters. A few examples of reports produced by this program would be:

- a. All work for Project No. 8601
- b. All work by Author Dalton
- c. All work for Lab SUYA
- d. All work for Initiator Champion

All of the management reports have been designed and written to be produced immediately upon request and to show at that time the latest status of any problem number issued by the Analysis and Simulation Branch at AFCRL. The ready access that management has to the Program Library Information File will aid in the future in the preparation of various statistical reports.

### 6.3 Statistical Reports

To date, only a few statistical reports have been produced because of the concentration of effort upon developing documentation standards and verification techniques and that of collecting data for the Program Library Information File. As the program library system steadily grows (there are presently 669 completed problem numbers encompassing over 1000 programs in the PLIF), a number of programs will be designed to produce statistical reports for management's use.

## SECTION 7

### CONVERSION

Toward the end of the contract, a CDC 6600 computer system became operational at AFCRL. It was necessary to convert the following programs from the IBM 7094/44 DCS System to the CDC 6600 system:

- LIBTAPE - Deck storage program
- LIBPUN - Deck retrieval program
- LIBABS - Tape listings of abstracts and source programs
- INSERT-ABS - Edit and update program for PLIF
- SMPG - Card listing program
- LIBALL - Listing of selected problem numbers
- LIBIND 1,2 - Key-word index program
- FILE - Pre-sort program for system SORT
- FILSORT - System sort-initiator, lab, problem no, project no.
- PRINT-SORT - Print program for FILSORT
- LIBCHS - Listing of selected information
- PRINTITLAB - Listing of title description
- RETPICK - Listing of selected files

Any program that had been written in machine language for the IBM 7094/44 DCS System was rewritten in Fortran IV prior to conversion to the CDC 6600.

Some of the major problems encountered in converting programs from the Direct Couple System to the CDC SCOPE Operating System were as follows:

#### 1. Word Size

Both the IBM 7000 series and the CDC 6600 computers are word-oriented machines; however, CDC 6600 word size is 60 bits long as opposed to the IBM 7000 series word size of 36 bits. This difference in word size causes some specific problems in shifting, masking and floating point formats.

##### A. Shifting

All shift functions written in Fortran for the DCS must be converted to the CDC shift functions or a logical procedure must be substituted to perform a comparable manipulation; and this modification must account for the larger word size.

##### B. Masking

All masking instructions written in FORTRAN on the DCS must be modified to conform with the CDC syntax. The actual masks themselves must also be modified to conform with the larger word size of the CDC 6600.



### C. Floating-Point Format

The floating-point format on the CDC 6600 is composed of a 48-bit integer coefficient with an 11-bit biased exponent. The IBM 7000 series computer uses a 27-bit integer coefficient with an 8-bit biased exponent. Thus the CDC will show more significance than the 7094. IBM floating point words on tape must be converted to conform to the CDC floating point format; consequently, routines were written in FORTRAN and COMPASS to do this conversion.

### 2. BCD Octal Code

The octal character code on the IBM 7000 series computers and the CDC 6600 are different; therefore, it is necessary to convert all IBM BCD characters to the equivalent CDC octal character code. When there is no corresponding character for the IBM code in CDC, a substitute must be made. All FORTRAN DCS Data statements which express BCD characters in their octal equivalents must be converted to the CDC octal character equivalent.

### 3. Control Card Language

All DCS IBSYS control cards must be replaced by CDC SCOPE control cards for the program to operate under the SCOPE Operating System. The SCOPE Operating System has an extensive control card language and can perform functions on-line which with the DCS had to be done off-line using the IBM 1460. One advantageous option which the SCOPE control card language gives the programmer is the ability to perform predefined utility routines if his program should terminate in an error mode. CDC uses a FORTRAN Program card at the beginning of the main FORTRAN routine which specifies all I/O devices used during execution, and sets up the logical and physical linkages for I/O units.

### 4. Non-Standard Tapes

The IBM 1460 was used to preprocess non-standard DCS tapes so they could be processed by the IBM 7094 II/7044 DCS. The IBM 1460 was used to pack the data into 36-bit words and to ensure correct parity for the DCS system. The CDC 6600 eliminates the need for the IBM 1460, and can process non-standard CDC tapes directly. Routines have been written for the CDC 6600 to read the following non-standard CDC tapes on the CDC 6600:

1. Library data base tapes created on a PDP-9
2. Library of Congress MARC II tapes created on an IBM 360/30
3. Unblocked IBM 7094 II/7044 DCS tapes

The data on these tapes also had to be converted to CDC-character and format codes. Routines were written to process locked IBM 7094 II/7044 DCS tapes utilizing the routine TCONVT.

#### 5. Machine Language

The DCS Machine Language (MAP) and the SCOPE Operating System Machine Language (COMPASS) are not compatible; i.e. MAP cannot be easily converted into COMPASS. It was found to be much easier to convert MAP routines into CDC FORTRAN during the conversion of AFCRL Research Library DCS Programs to the CDC 6600. The use of CDC FORTRAN is possible in these conversions because of the availability of functions in CDC FORTRAN to perform the functions that were necessary in MAP on the IBM 7094.

#### 6. Overlays

A program can be divided into parts for an overlay when it exceeds available memory if each part is independent and can be called and executed as necessary. Each part (overlay) consists of a single main program and any necessary subprograms. Differences in control cards, END-OF-FILE recognition in Input/Output routines, labelled COMMON statements and item 1- through 5 mentioned previously all need to be considered when converting from the IBM 7000 series to the CDC 6600.

## SECTION 8

### CONTRIBUTORS

The writers express their thanks to Miss Eunice C. Cronin, Analysis and Simulation Branch (SUYA), AFCRL Computation Center, Air Force Cambridge Research Laboratories, and to Mr. Austin A. Almon, Jr., Contract Monitor, whose technical guidance and experience were invaluable in the preparation of this report.

APPENDIX A

SAMPLE DOCUMENTATION FORM  
UTILIZING THE IBM 7094 II/7044 DCS

Program Name: \_\_\_\_\_

Prepared under Contract No. \_\_\_\_\_ for

Air Force Cambridge Research Laboratories (AFSC)  
Analysis & Simulation Branch

Contract Monitor: \_\_\_\_\_

Problem Number: \_\_\_\_\_

Project Number: \_\_\_\_\_

Date: \_\_\_\_\_

Preceding page blank

Date

Prepared By: \_\_\_\_\_

Reviewed By: \_\_\_\_\_

Approved By: \_\_\_\_\_

## TABLE OF CONTENTS

SECTION	CONTENTS	PAGE
I	List of Illustrations	1.0
II	Introduction	2.0
III	Functional Description	3.0
IV	Hardware Requirements	4.0
V	Software Requirements	5.0
VI	Input	6.0
	Magnetic Tape	6.1
	Input Cards	6.2
	Other Media	6.3
VII	Output	7.0
VIII	Mathematical or Logical Procedures	8.0
IX	Error Routines and Indications	9.0
X	Program Restrictions and Timing	10.0
XI	Operating Instructions	11.0
XII	Verification of Operation	12.0

1.0

I. LIST OF ILLUSTRATIONS

LIST

PAGE NO.

Note: Illustrations may appear anywhere in text or grouped together before appendix.



## II. INTRODUCTION

Please be concise and not over one page in length. Should cover these topics, when applicable:

1. Any relevant background on the need for program
2. What program does, not how it is done
3. Possible expansion, modification or other uses for the program
4. Other relevant information such as program limitations, timing limitations, environment limitations, etc.

3.0

III. FUNCTIONAL DESCRIPTION

General explanation of what program does and how it is done. Something about input and output of program and action taken by program. Attach FLOW-CHART and PROGRAM LEGEND (when applicable.)

IV. HARDWARE REQUIREMENTS

List special equipment features.

Source Computer Configuration

Object Computer Configuration

Minimum and Maximum Memory Requirements:

5.0

V. SOFTWARE REQUIREMENTS

Source Program Language: \_\_\_\_\_

Prerequisite Programs and Subroutines

Name

Reference or  
SUYA Program No.

6.0

VI. INPUT

VII. OUTPUT

## A. Standard:

Describe the normal program printed output and the configuration of any magnetic, paper tape and/or any punched cards generated.

## B. Error:

Message

Reason

**VIII. MATHEMATICAL OR LOGICAL PROCEDURES**

(Include: Mathematical Symbols Used, Mathematical Techniques, Equations, Special Programming Techniques, Table Structures, Indexing and Indirect Addressing, Initialization and Re-initialization.)

**IX. ERROR ROUTINES AND INDICATIONS**

<u>Error Condition</u> <u>Checks</u>	<u>Internal &amp; External</u> <u>Indications</u>	<u>Recovery</u> <u>Procedure</u> <u>or</u> <u>Action</u> <u>Required</u>
---	--	--

This part of the document should cover all error condition checks, indications and recovery procedures. Each leading to a possible error, such as invalid data, sequence errors, incorrect formats, etc., should be listed. The document should define what internal and external indicators are set and what messages are printed out, and what action is required for each error message.



#### X. PROGRAM RESTRICTIONS AND TIMING

List here any special program restrictions that apply to this program only. If program restrictions are given elsewhere in the text they should be repeated here.

Such items as maximum and minimum allowable size of input and output values, special buffer areas, etc., or anything that is unique to this program only should be listed here.

#### Timing

List here the running time for the program.

(in terms of number of records processed, number of cards processed, lines of output, or some such meaningful, measureable quantity.)

**XI. OPERATION INSTRUCTIONS****Control Cards****DCS Input Card****Yes** \_\_\_\_\_**No** \_\_\_\_\_**Sense Switch Setting and Functions****Tape Units Used****Error Halts****Indicators****Restart Procedure****End of Job Indication****Size of Printout Paper****No. of Copies**

12.0

**XII. VERIFICATION OF OPERATION**

**List of Procedures for Test to Verify Program Operation:**

**(Data and Output in Appendix)**

APPENDIX B

Proposed  
Sample Documentation Form  
Utilizing the CDC 6600

Program Name \_\_\_\_\_

Prepared under Contract No. \_\_\_\_\_ for

Air Force Cambridge Research Laboratories (AFSC)  
Analysis and Simulation Branch

Contract Monitor: \_\_\_\_\_

Problem Number: \_\_\_\_\_

Project Number: \_\_\_\_\_

Date: \_\_\_\_\_

***Preceding page blank***

Task Requested by: \_\_\_\_\_

Lab: \_\_\_\_\_ Telephone: \_\_\_\_\_ Date: \_\_\_\_\_

Task Performed by: \_\_\_\_\_

Company: \_\_\_\_\_ Telephone: \_\_\_\_\_ Date: \_\_\_\_\_

Task Reviewed by: \_\_\_\_\_

Telephone: \_\_\_\_\_ Date: \_\_\_\_\_

Task Approved by: \_\_\_\_\_

Telephone: \_\_\_\_\_ Date: \_\_\_\_\_

## TABLE OF CONTENTS

SECTION	CONTENTS	PAGE
I	Documentation Report Information	1.0
II	Revision	2.0
III	List of Illustrations	3.0
IV	Introduction	4.0
V	Functional Description	5.0
VI	Hardware Requirements	6.0
VII	Software Requirements	7.0
VIII	Input	8.0
IX	Output	9.0
X	Mathematical or Logical Procedures	10.0
XI	Error Routines and Indications	11.0
XII	Program Restrictions and Timing	12.0
XIII	Instructions for Job Setup	13.0
XIV	Verification of Operation	14.0

I. DOCUMENTATION REPORT INFORMATION

Totally New Effort	Yes	____	No	____
Revision	Yes	____	No	____
Program Conversion	Yes	____	No	____
Production Running	Yes	____	No	____
Decks Available	Yes	____	No	____
Assembly Listing	Yes	____	No	____
Test Case Results	Yes	____	No	____
Dayfile Included	Yes	____	No	____
Analysis Only	Yes	____	No	____

Results Used in Scientific Report? Yes \_\_\_\_ No \_\_\_\_

If yes, enter report name and ID on next line.

---

Use the following available space for any additional information pertinent to this report.



II. REVISION

Revision Number: \_\_\_\_\_

Revision to Problem Number \_\_\_\_\_

Revision Date: \_\_\_\_\_

Documentation Form Pages Affected by this Revision: \_\_\_\_\_

III. LIST OF ILLUSTRATIONS

LIST

PAGE NO.

Note: Illustrations may appear anywhere in text or grouped together before Appendix.

IV. INTRODUCTION

Please be concise and not over one page in length. Should cover these topics, when applicable:

- 1) Any relevant background on the need for program
- 2) What program does, not how it is done
- 3) Possible expansion, modification or other uses for the program
- 4) Other relevant information, such as program limitations, timing limitations, environment limitations, etc.

V. FUNCTIONAL DESCRIPTION

General explanation of what program does and how it is done. Something about input and output of program and action taken by program. Attach FLOW CHART and PROGRAM LEGEND when applicable.

VI. HARDWARE REQUIREMENTS

Computer Type \_\_\_\_\_

## Computer Requirements:

Central Processor (CP)      Yes \_\_\_\_\_ No \_\_\_\_\_

Peripheral Processor (PP)      Yes \_\_\_\_\_ No \_\_\_\_\_

Plotter      Yes \_\_\_\_\_ No \_\_\_\_\_

Calcomp \_\_\_\_\_

CRT \_\_\_\_\_

Other (specify) \_\_\_\_\_

Printer      Yes \_\_\_\_\_ No \_\_\_\_\_

Punch      Yes \_\_\_\_\_ No \_\_\_\_\_

Graphics      Yes \_\_\_\_\_ No \_\_\_\_\_

Disk Packs      Yes \_\_\_\_\_ No \_\_\_\_\_

Common \_\_\_\_\_

Private \_\_\_\_\_

VII. SOFTWARE REQUIREMENTS

## Operating System Software:

SCOPE \_\_\_\_\_

IBSYS \_\_\_\_\_

Other (specify) \_\_\_\_\_

## Compiler:

FTN \_\_\_\_\_

RUN \_\_\_\_\_

Other (specify) \_\_\_\_\_

## Source Program Language:

FORTRAN IV \_\_\_\_\_

COMPASS \_\_\_\_\_

Other (specify) \_\_\_\_\_

## Supporting System Software:

Sort/Merge \_\_\_\_\_

Update \_\_\_\_\_

Utility \_\_\_\_\_

Editsym \_\_\_\_\_

Other (specify) \_\_\_\_\_

VIII. INPUTA. Punched Card

<u>Card</u> <u>No.</u>	<u>Variable</u> <u>Name</u>	<u>Card</u> <u>Col.</u>	<u>Format</u>	<u>Variable Description</u>
---------------------------	--------------------------------	----------------------------	---------------	-----------------------------

B. Magnetic Tape

Name \_\_\_\_\_; No. of Files \_\_\_\_\_; Density \_\_\_\_\_ BPI; No. of Tracks \_\_\_\_\_;  
 Type (Bin. or BCD) \_\_\_\_\_; Label: Yes \_\_\_\_\_ No \_\_\_\_\_; No. of Reels \_\_\_\_\_.

Number of Words in Record \_\_\_\_\_

Indicate record format and attach separate sheet as illustration.

C. Other

Graphics \_\_\_\_\_ Paper Tape \_\_\_\_\_ Keyboard \_\_\_\_\_ Disk Pack \_\_\_\_\_

IX. OUTPUT

Describe the normal program printed output and the configuration of any magnetic tape, paper tape, and/or any punched cards generated.

Page Estimate = \_\_\_\_\_

Size of Printout Paper \_\_\_\_\_

No. of Copies \_\_\_\_\_



2. MATHEMATICAL OR LOGICAL PROCEDURES

Describe logical steps taken by program to accomplish tasks outlined in the functional description section. Include mathematical symbols used, mathematical techniques, equations, special programming techniques, table structures, indexing and indirect addressing, initialization and re-initialization.

XI. ERROR ROUTINES AND INDICATIONS

<u>Error Condition</u> <u>Checks</u>	<u>Internal and External</u> <u>Indications</u>	<u>Recovery Procedure or</u> <u>Action Required</u>
---	--	--

Cover all error condition checks, indications and recovery procedures set up in your program. Anything leading to a possible error, such as invaled data, sequence errors, incorrect formats, etc., should be listed. The document should define what internal and external indicators are set and what messages are printed out, and what action is required for each error message.

## XII. PROGRAM RESTRICTIONS AND TIMING

List any special program restrictions that apply to this program only. If program restrictions are given elsewhere in the text, they should be repeated here.

Such items as maximum and minimum allowable size of input and output values, special buffer areas, etc., or anything that is unique to this program only should be listed here.

### Timing

List the running time for the program in terms of number of records processed, number of cards processed, lines of output, or some other meaningful, measurable quantity.

XIII. INSTRUCTIONS FOR JOB SETUP

List System Control Cards

Magnetic Tapes

Name	CC	OS	S or L	Unit	Description
_____	_____	_____	_____	_____	
_____	_____	_____	_____	_____	
_____	_____	_____	_____	_____	
_____	_____	_____	_____	_____	
_____	_____	_____	_____	_____	
_____	_____	_____	_____	_____	

Maximum Core Storage Needed for:

Compilation \_\_\_\_\_ octal

Execution \_\_\_\_\_ octal

Standard Test Case Time Allocation \_\_\_\_\_ seconds

Sense Switch Settings \_\_\_\_\_

XIV. VERIFICATION OF OPERATION

Rerun program and check results with  
sample output submitted.

Yes \_\_\_\_ No \_\_\_\_

Rerun program and compare tape(s)  
generated with original.

Yes \_\_\_\_ No \_\_\_\_

Other (specify)